The transmit inhibit signal indicated to radio **502-1** and, more particularly, to transmitter **705-1**, ultimately controls the signal radiated by radio **502-1**. In order to suppress radiation of a signal, it might be necessary to turn off or turn low the power amplifier and the RF/IF sections of transmitter **705-1**, as described earlier. It will be clear to those skilled in the art how to suppress output from transmitter **705-1**.

Setting the transmit inhibit signal prevents the remainder of frame **912** from reaching antenna unit **504**, as shown by frame **922**. When transmitter **705-2** completes lower latency-tolerant packet **951**, the transmit inhibit signal resets low, thereby allowing input to transmitter **705-1** to once again reach antenna unit **504**. The transmit inhibit signal, in combination with any intermediate logic gates required to format the control signal actually provided to transmitter **705-1**, acts as a preemption signal that effectively suppresses output from transmitter **705-1** during transmitter **705-2**'s transmissions, thereby avoiding interference.

Meanwhile, transmitter **705-1**, unaware that frame **912** did not fully reach antenna unit **504**, waits for an acknowledgement in accordance with automatic repeat request (ARQ) error correction, as is well understood in the art. Since frame **912** was effectively interrupted, transmitter **705-1** does not receive such an acknowledgement, and, after a timeout in accordance with the protocol, retries frame **912** (in the form of frame **913**.) As illustrated in FIG. **9**, as long as Bluetooth packet **951** is kept sufficiently short, transmitter **705-1** is no longer suppressed by transmitter **705-2** when transmitting frame **913**. Consequently, frame **913** in its entirety reaches antenna unit **504** (shown by frame **923**), and receiver **704-1** subsequently receives acknowledgement **932**. Recalling the 802.11/Bluetooth nature of the example depicted by FIG. **9**, the IEEE 802.11 ARQ error correction thus automatically compensates for sufficiently-short Bluetooth interruptions (i.e., interruptions that are not "fatal") without any changes to the protocols.

It will be clear to those skilled in the art that ARQ error correction will also automatically compensate for sufficiently-short transmissions from transmitter **705-2** of radio **502-2** that overlap receiver **704-1**'s receiving of data. In addition, it will be clear to those skilled in the art how to make and use alternative embodiments of the present invention for protocols that use other methods of error correction (e.g., forward error correction, etc.) In the case of forward error correction, for example, the interruption of a transmission is not fatal as long as the interruption is kept short enough so that the number of suppressed bits is below the particular error correction threshold.

So far throughout the exemplary sequence depicted in FIG. **9**, radio **502-1** has been active, as shown by the "low" value of signal **904**, corresponding to the first idle indication signal of radio **502-1**, which is provided by signaling link **508-1** to radio **502-2**. After acknowledgement frame **932**, radio **502-1** enters power-save (i.e., idle) mode, as shown in FIG. **9** by the transition of first idle indication signal (signal **904**) from low to high. Transmitter **705-2**, upon detecting this transition, takes advantage of this situation by transmitting higher latency-tolerant packet **952** (e.g., an asynchronous connection-less [ACL] packet, etc.). Thus, instead of preempting transmitter **705-1**, as is done for transmissions with a lower latency tolerance (e.g., transmission **951**, etc.), transmitter **705-2** waits for radio **502-1** to enter power-save mode before initiating transmissions with a higher latency tolerance (e.g., **952**, etc.).

When radio **502-1** exits power-save mode (i.e., "wakes up"), it executes a "warm-up sequence" before transmitting any frames, as is well known in the art. If radio **502-1** happens to wake up while transmitter **705-2** is still transmitting, radio **502-2**, which detects that radio **502-1** has awakened, terminates transmitter **705-2**'s transmissions. As will be clear to those skilled in the art, the warm-up sequence of radio **502-1**, operating in the example in accordance with the Bluetooth protocol, gives transmitter **705-2** plenty of time to gracefully terminate any in-progress transmissions. Any "left-over" information that transmitter **705-2** was unable to transmit before radio **502-1** awoke is queued for the next time that radio **502-1** enters power-save mode; this postponement is not problematic since, by definition, the information has a higher latency tolerance. If, instead, this information had a lower latency tolerance, transmitter **705-2** would have previously preempted transmitter **705-1**, as described above;

FIG. **10** depicts a block diagram of radio **502-1** in another variation of the third illustrative embodiment of the present invention. FIG. **10** is similar to FIG. **7**, except that the signaling links between radios **502-1** and **502-2** are interfaced directly to multi-radio host interface **1002**. Consequently, channel-access controller **1001**, multi-radio host interface **1002**, and path **1005** are different from channel-access controller **701**, multi-radio host interface **702**, and path **705**, respectively.

Channel-access controller **1001** provides the medium access control functionality for communicating in accordance with a first air interface (e.g., 802.11, Bluetooth, etc.). In this regard, it provides the same functionality as channel-access controller **701**. It accepts host data from multi-radio host interface **1002** via path **1005**. It provides data from host **501** to baseband controller **703** via path **712** for preparation for transmission. Channel-access controller **1001** also provides data received over the air from baseband controller **703** via path **712** to host **501** through path **1005** and multi-radio host interface **1002**. Channel-access controller **1001** can track whether it has control or radio **502-2** has control of the communications band at any given moment. Consequently, channel-access controller **1001** can control antenna switching at antenna switch **503** via path **511-1**. Alternatively, channel-access controller **1001** can operate uninformed of the status of radio **502-2**.

Channel access controller **1001** can pass to radio **502-2** via signaling link **508-1** information representative of receiver **704-1** and transmitter **705-1**, received through path **1006**. Channel access controller **1001** can pass to receiver **704-1** and transmitter **705-1** via path **1006** information representative of radio **502-2**, received through signaling link **508-2**. It will be clear to those skilled in the art how to make and use channel-access controller **1001**.

In accordance with the illustrative embodiment of the present invention, multi-radio host interface **1002** provides the interface between host **501** and radio **502-1**. Multi-radio host interface **1002** accepts data blocks from host **501** via host data link **506**. Multi-radio host interface **1002** then determines whether it should (1) transfer each data block to channel-access controller **1001** via path **1005**, if the data block is meant for radio **502-1**, or (2) relay the data block over to radio **502-2** via link collateral radio data link **507**. Multi-radio host interface **1002** accepts data blocks from channel-access controller **1001** and transfers them to host **501**. In other words, multi-radio host interface **1002** provides multiple logical channel interfaces on a single physical channel interface to host **501**. After reading this specification, it will be clear to those skilled in the art how to make and use multi-radio host interface **1002**.

Multi-radio host interface **1002** terminates one end of collateral radio data link **507**, as well as signaling links